# Integration of an Informatics System in a High Throughput Experimentation. Description of a Global Framework Illustrated Through Several Examples.

Benoît Celse, Stéphane Rebours, Fabrice Gay, Pauline Coste, Loïc Bourgeois, Olivier Zammit, Vassilissa Lebacque

*Oil & Gas Science and Technology – Rev. IFP Energies nouvelles*, Vol. 68 (2013), No. 3, pp. 445-468

DOI: 10.2516/ogst/2013109

**Dossier**

*Discovery and Optimization of Catalysis and Solvents for Absorption Using High Throughput Experimentation*
Découverte et optimisation de catalyseurs et d'absorbants par expérimentation haut débit

# Integration of an Informatics System in a High Throughput Experimentation. Description of a Global Framework Illustrated Through Several Examples

B. Celse[1]*, S. Rebours[1], F. Gay[2], P. Coste[2], L. Bourgeois[2], O. Zammit[2] and V. Lebacque[3]

[1] *IFP Energies nouvelles, Rond-point de l'échangeur de Solaize, BP 3, 69360 Solaize - France*
[2] *Tech Advantage, Rond-point de l'échangeur de Solaize, BP 3, 69360 Solaize - France*
[3] *Laboratoire G-SCOP, 46 avenue Félix Viallet, 38031 Grenoble Cedex 1 - France*
*e-mail: benoit.celse@ifpen.fr - stephane.rebours@ifpen.fr - fabrice.gay@tech-advantage.com - pauline.coste@tech-advantage.com loic.bourgeois@hotmail.fr olivier_zammit@yahoo.fr - vlebacque@gmail.com*

* Corresponding author

**Résumé** — **Intégration informatique des outils d'expérimentation haut débit. Présentation d'une architecture globale *via* plusieurs exemples** — L'Expérimentation Haut Débit (EHD) est un domaine en plein essor. Cependant, les gains de productivité obtenus *via* la synthèse ou le test parallélisé de catalyseurs peuvent être annihilés par une mauvaise gestion de données (nombreuses saisies manuelles, difficulté d'accès à l'information, etc.). Dans ce document, une nouvelle architecture permettant d'intégrer les unités EHD dans un système d'information global est présentée. Des outils informatiques dédiés ont été développés. Ils permettent des gains de temps spectaculaires dans la conduite des unités EHD, le stockage des données et l'extraction rapide des informations pertinentes.

L'approche retenue a été guidée par une approche Agile (Agile Alliance (2012) http://www.agilealliance.org/the-alliance/) [1] basée sur une très forte collaboration entre les chimistes et les informaticiens. Excel, l'outil principal des chimistes, a été positionné au cœur du système d'information avec des liens bidirectionnels (entrée/sortie) avec les bases de données et les différentes unités pilotes.

Plutôt qu'une description globale du système d'information qui serait longue et fastidieuse, le *framework* est présenté *via* 3 exemples principaux :

– planification de données en utilisant des outils de gestion de production ;
– gestion optimisée des données (stockage, requêtes), analyse de données ;
– exemples d'application sur des unités pilotes : développements d'interfaces dédiées pour la conduite, le monitoring des unités et l'exploitation multi *runs* et multi unités pilotes.

***Abstract* — *Integration of an Informatics System in a High Throughput Experimentation. Description of a Global Framework Illustrated Through Several Examples*** — *High Throughput Experimentation (HTE) is a rapidly expanding field. However, the productivity gains obtained via the synthesis or parallel testing of catalysts may be lost due to poor data management (numerous manual inputs, information difficult to access, etc.). A global framework has then been developed. It includes the HTE pilot plants in the global information system. It produces dedicated computer tools offering spectacular time savings in the operation of HTE units, information storage and rapid*

*extraction of relevant information. To optimize the productivity of engineers, Excel has been included in the system by adding specific features in order to treat it as an industrial tool (development of additional modules, update of modules, etc.).*

*The success obtained by setting up the information system is largely due to the chosen development method. An Agile method (Agile Alliance (2012) http://www.agilealliance.org/the-alliance/)[1] was chosen since close collaboration between the computer specialists and the chemist engineers is essential.*

*Rather than a global and precise description of the framework which might be boring and tedious, the global framework is presented through 3 examples:*

- *scheduling experiments applied to zeolite synthesis;*
- *data management (storage and access);*
- *real application to pilot plant: dedicated interfaces to pilot and supervise HTE pilot plants, comparison of tests runs coming from several pilot plants.*

## INTRODUCTION

High throughput, combinatorial synthesis and screening techniques are increasingly utilized in materials science research. High Throughput Experimentation (HTE) places particular engineering demands on instrument design: in order to operate efficiently, extensive device automation is required and an integrated approach to data management must be adopted.

High throughput screening techniques have the potential to generate large volumes of data both directly from measurements performed during screening and indirectly from the metadata arising from the initial sample synthesis and later processes. These metadata are essential for maintaining the provenance library samples over their life cycle. Thus, a critical component of a HTE system is an information system for managing data and preserving the relationships between them.

In essence, an information system provides the infrastructure for data collection, storage, processing and presentation. Desirable qualities for the design of information system for combinatorial research are variously discussed in [2-6]. Although commercial software is available (IDBS e-workshop, Accelrys Pipeline pilot, etc.), both for data management (IDBS e-workbook, Accelrys pipeline pilot, StoCat, etc.) [7] and device control [8], it is frequently costly and often additional work is required to integrate it with specific instrumentation. Consequently, local solutions are often developed [9, 10], although these are typically customized for a particular application and may suffer from a lack of generality.

A global framework has been developed in order to include HTE pilot plant in the global information system. It is based on the main tools of the chemical engineer: Excel.

Rather than a global and precise description of the framework which might be boring and tedious, the global framework is presented through 3 examples:

- Section 1 explains development methodology chosen in order to improve productivity;
- Section 2 illustrates the use of scheduling techniques in order to optimize the use of pilot plants. Although these techniques are in fact rarely used in HTE, they are highly efficient;
- Section 3 describes data handling (storage, query and multi-dimensional analysis). The number of data handled is relatively large (several tens of millions). Safe and secure storage must therefore be planned, not forgetting the reporting aspects;
- Section 4 describes practical examples of automation (graphical user interface base on Excel), monitoring and multi-runs analyses.

## 1 DEVELOPMENT METHODOLOGY

Agile methodology [1] is an alternative to traditional project management, typically used in software development. It helps teams respond to unpredictability through incremental, iterative work cadences, known as sprints. Agile methodologies are an alternative to waterfall, or traditional sequential development.

In 1970, Royce presented a paper entitled "Managing the Development of Large Software Systems", which criticized sequential development. His presentation asserted that a project could not be developed like an automobile on an assembly line, in which each piece is added in sequential phases. In such sequential phases, every phase of the project must be completed before the next phase can begin. Royce recommended against the phase based approach in which developers first gather all of a project's requirements, then complete all of its architecture and design, then write all of the code, and so on. Royce specifically objected to this approach due to the lack of communication between the specialized groups that complete each phase of work.

It is easy to see how the above methodology is far from optimized compared to agile methodology. First of all, it assumes that every requirement of the project can be identified before any design or coding occurs. Put another way, do you think you could tell a team of developers everything that needed to be in a piece of software before it was up and running? Or would it be easier to describe your vision to the team if you could react to functional software? Many software developers have learned the answer to that question the hard way: at the end of a project, a team might have built the software it was asked to build but, in the time it took to create, business realities have changed so dramatically that the product is irrelevant. In that scenario, a company has spent time and money to create software that no one wants. Could not it have been possible to ensure the end product would still be relevant before it was actually finished?

Agile development methodology provides opportunities to assess the direction of a project throughout the development lifecycle. This is achieved through regular cadences of work, known as sprints or iterations, at the end of which teams must present a potentially shippable product increment. By focusing on the repetition of abbreviated work cycles as well as the functional product they yield, agile methodology is described as "iterative" and "incremental". In waterfall, development teams only have one chance to get each aspect of a project right. In an agile paradigm, every aspect of development – requirements, design, etc. – is continually revisited throughout the lifecycle. When a team stops and re-evaluates the direction of a project every two weeks, there is always time to steer it in another direction.

The results of this "inspect-and-adapt" approach to development greatly reduce both development costs and time to market. Because teams can develop software at the same time they are gathering requirements, the phenomenon known as "analysis paralysis" is less likely to impede a team from making progress. And because a team's work cycle is limited to two weeks, it gives stakeholders recurring opportunities to calibrate releases for success in the real world. Agile development methodology helps companies build the right product. Instead of committing to market a piece of software that has not even been written yet, agile empowers teams to continuously re-schedule their release to optimize its value throughout development, allowing them to be as competitive as possible in the marketplace. Development using an agile methodology preserves a product's critical market relevance and ensures a team's work does not wind up on a shelf, never released. This is clearly an attractive option for stakeholders and developers alike.

Agile method is based on following principles:

- our highest priority is to satisfy the customer through early and continuous delivery of valuable software;
- welcome changing requirements, even late in development. Agile processes harness change for the customer's competitive advantage;
- deliver working software frequently, from a couple of weeks to a couple of months, with a preference to the shorter timescale;
- business people and developers must work together daily throughout the project;
- build projects around motivated individuals. Give them the environment and support they need, and trust them to get the job done;
- the most efficient and effective method of conveying information to and within a development team is face-to-face conversation;
- working software is the primary measure of progress;
- agile processes promote sustainable development. The sponsors, developers, and users should be able to maintain a constant pace indefinitely;
- continuous attention to technical excellence and good design enhances agility;
- simplicity – the art of maximizing the amount of work not done – is essential;
- the best architectures, requirements, and designs emerge from self-organizing teams.

At regular intervals, the team reflects on how to become more effective, then tunes and adjusts its behaviour accordingly.

## 2 EXPERIMENT PLANNING

This paragraph gives an example of how scheduling techniques (coming from production management) are used for experiment planning. It has been applied for the parallel synthesis of zeolites. Its objective is the discovery of new zeolite phases, while reducing to a minimum the factors "time" and "amount of reactive".

These synthesis are difficult to manage because of the time required which might be between 3 to 21 days for each sample. We then decided to carry out parallel experiments. This paragraph describes the computer science tool developed to tackle this problem.

To identify XRD signal, a in-house software (ZeoStat similar to PolySnap) and a commercial software FPMatch© (provided by Socabim) was used. FPMatch© uses weighted least squares method. It was tested against PolySnap© and proved, in our cases, more efficient. A dedicated in house tool was developed (VisuDRX) in order to compare and visualize XRD diagrams.

The methodology in order to deal with data (XRD automatic determination, data analysis, etc.) is not described in this paper because it is well described in other papers [11-16]. In this paper, we prefer to focus on one point which is few described in high throughput experiments: experimental planning.

Besides, we will not describe Design of Experiments (DOE) technique in this paper. Interested readers can read following references [17-22].

### 2.1 Description of the Problem

The solution to this problem involves developing software to determine the optimum conditions required to synthesise zeolites. The problem consists in performing tasks which each correspond to a set of experiments likely to discover new zeolite phases. The experiments to be performed for each task are not all known in advance. A first series of experiments (phase I experiments) is conducted to determine the optimum duration for each product. It is based on the metastable nature of zeolite structure. Indeed, obtaining amorphous phase may mean that too short synthesis time was applied, on the other hand, if dense phases are obtained it may be that a micro porous structure has evolved. Different operating conditions (screening of the gel composition) are then tested (phase II) with the selected duration to improve the product quality (amount of reactive). This corresponds to a second series of experiments.

A set of racks (containing 10 samples) are available to conduct the experiments. This tool was supplied by the company *TopIndustrie* and *TECAN* for the robotic arm distribution. Several experiments of equal duration and temperature can be conducted simultaneously in each rack (it is not possible to take one sample from a rack). The experiments last between 3 and 21 days. Several thousand experiments will be conducted.

There is a two-fold objective: firstly, maximise filling of the racks in order to complete all experiments as quickly as possible; secondly, finish the tasks regularly and as quickly as possible to determine the quality of the products obtained (amount of reactive).

The actions to be performed on a rack, in addition to the experiments, can be broken down into elementary operations (washing, filling, etc.) of fixed duration. Some of these operations require operator intervention. The planning must take into account operator days off, possible breakdowns and stock outages.

#### 2.1.1 Sequence of Operations

We start with a certain chemical substance $S$ (for structurant), which will be exposed to a series of heating procedures of various durations (from 3 to 21 days). From their past experience, the chemists know very well all durations, which possibly might be useful in connection with the given chemical $S$. But the chemists do not know in advance which durations will finally be used during the experimentations with $S$. More precisely, the chemists have for $S$ a virtual pool of heating durations that may be presented in form of an out-tree (*Fig. 1*). The root and all other nodes of the out-tree correspond to chemical $S$. The arcs indicate the operations and the labels are the heating durations that $S$ will be exposed to. Once an operation is finished, the chemists will analyse the substance (using X Ray Diffraction) and decide what to do next (depending on the amount of crystallised phase), which means that they will choose the following operation and its heating duration. In this way, starting with $S$, the sequence of operations is forming a path in the out-tree ($p_2$, $p_5$, $p_8$, $p_{10}$ in *Fig. 1*). The actual path is unknown in advance. Nevertheless, the chemists have certain information about the possible subsequent operations (for instance, the possible successors of $p_5$ (7 days) are $p_8$ (3 days) or $p_9$ (14 days)). We are, therefore, confronted with a so-called semi-online scheduling problem.

In the example, the path terminates with duration $p_{10}$, which is found by the chemists to be the appropriate

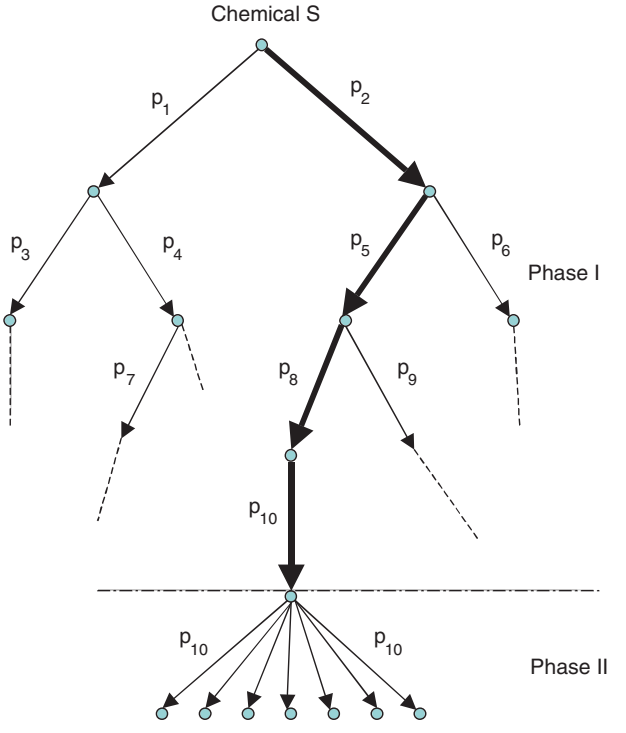


Figure 1

Out-tree of operations.

(optimal) heating exposure for chemical $S$. This is the end of the Phase I experiments. In Phase II, further experiments are carried out with the optimal duration (here $p_{10}$) but under various experimental conditions in order to improve even more the quality of the chemical. Note that the Phase II operations can be done in parallel.

#### 2.1.2 Resources and Objectives

A computerized planning of several hundreds operations is needed. The individual heating durations last from 3 days to 21 days so that the total series of experiments with different chemicals might take a whole year.

The samples are placed into a rack, which possesses a total of $c$ reactors (10 in our case). This means that the rack can be filled with a maximum of $c$ samples to be treated at the same time. In terms of the scheduling theory, the $c$ samples form a parallel batch (see [23, 24]). In addition, the chemicals of the same batch have to be batch-compatible [25], which means in our case that they will be exposed to exactly the same heating duration. There are $m$ such racks and the heating device is a type of oven.

Between the different heating operations, the rack has to be prepared. A transition and setup time is required and we have the following sequence of operations:

$$\text{filling} \Rightarrow \text{heating} \Rightarrow \text{filtering} \Rightarrow \text{rinsing}$$

After the rinsing procedure, the rack is again ready for the filling with a chemical. The heating operation cannot be interrupted. However, the much shorter secondary operations (filling, filtering, rinsing) may be started for instance before a holiday and resumed and finished afterwards.

There is a further interesting aspect, which does not exist in the classical scheduling literature. After the heating procedure, a chemist (an operator) has to be present for the sensitive handling of the material and the setup for the next experiment. Since the chemists are not always available due to breaks, holidays, weekends, etc., we have to respect in our schedules these operator non-availability periods (Ona periods). In [26-28], the theory of scheduling with such Ona periods has been recently have investigated. Note that this is different from the well-established concept of machine non-availability periods [29, 30].

We have several objectives to consider for planning the chemical experiments. Of course, the chemists would like to finish everything as early as possible. This corresponds to the makespan criterion $C_{\max}$. We also want to terminate evenly all experiments, conducted on a given chemical, so that the chemists obtain regularly results on their quality, which would also balance their work during the time of the experiments. A third objective concerns the utilization of the racks. The chemists like to fill the racks as much as possible, close to their capacity $c$.

#### 2.1.3 Scheduling Model

We can now translate our planning problem into the language of scheduling theory. The chemical substances to examine, $S_j$ for $j = 1, 2, \ldots, n$, correspond to the jobs to be scheduled. Each of these jobs consists of a sequence of operations (a path in the out-tree), where the next operation and its duration is only known after having observed the outcome of its predecessor (semi-online scheduling). After the sequential operations (Phase I), independent operations, all of which have the same duration given by the preceding operation, are to be processed (Phase II).

We have a parallel machine system, consisting of $m$ identical machines (the racks). The machines are so-called batch-machines, which can handle simultaneously up to $c$ operations of equal duration.

There are several performance criteria for the schedules that should be incorporated. Let us denote by $C_j$ the completion time of job $S_j$, *i.e.* the instant where all its Phase II operations are finished. Then we consider the following criteria:

$$\text{total completion time } C_{\max} = \max\left(C_j\right) \quad (1)$$

$$\text{sum of completion times } \sum C_j \quad (2)$$

$$\text{sum of resource utilization times } \sum R_i \quad (3)$$

The exact definition of the quantity $R_i$ in (3) will be given in Section 2.2.1. The operations are not preemptable (no interruptions), whereas the setups may be pre-empted. There are further constraints, for instance one has to satisfy Ona periods, where the operator is not available. However, human intervention is necessary at the end of an operation. The scheduling problem is NP-hard in terms of the complexity theory, since minimizing $C_{\max}$ is already NP-hard on two parallel machines, even without any additional resources [31].

### 2.2 Solution Approach

The scheduling problem is rather complicated and of large size. It will not be possible to solve it to optimality in a reasonable time. To respect the semi-online character of the problem, we shall develop an algorithm that readjusts the currently proposed solution, each time a new event (filling, break-down) occurs. The method is

decomposed into two parts: building the initial (static) schedule, followed by the main (dynamic) schedule.

#### 2.2.1 Initial Schedule

The aim of the initialisation is to generate a sufficient number of operations in order to fill all racks in the schedule that will follow and also to get all chemicals (jobs) under way. Here, we deal with the start of Phase I operations, when only the first experiments of the different chemicals to be tested are available. This corresponds to the roots of the out-trees for the different chemicals and eventually one of their possible successors. Since for each chemical the chemists have knowledge, by their past experience, of the most probable path in the out-tree of possible operations, the problem is essentially a static scheduling problem. In practice, we had about 20 different chemicals to analyse under different experimental parameters, which represent a set of 200 operations to be scheduled. There are also some precedence constraints between the jobs to satisfy. In addition, since we are at the root of the out-trees, only a very limited number of different durations is possible. For these reasons, corresponding to the actual industrial setting, we consider that all racks can be completely filled. Thus we disregard the batching problem in this step and we only consider the schedule of constituted batches filled to capacity on parallel machines (the racks), with the additional operator availability constraints described earlier.

As a possible solution approach, we first tried constraint propagation software. But it took several hours of computation time without obtaining a first feasible solution. We then turned with more success to integer programming software (CPLEX under OPL Studio).

The objective function is defined in form of a weighted combination of criteria (1) and (3):

$$\alpha C_{\max} + \beta \sum R_i \qquad (4)$$

where $R_i$ represents the time where rack $i$ has completed all the batches assigned to it. Since the time horizon of the initialisation is short, we chose a discrete time formulation (half day periods) for the problem. The main binary decision variables are then $r_{jt}$, which are equal to 1 if batch $j$ starts its processing at time $t$, and 0 otherwise. This formulation has the advantage to handle quite easily the special unavailability periods of the problem. One can simply precompute all the instants when it is forbidden to start batch $j$, and add the constraint that $\sum_t r_{jt} = 0$ on all such instants.

Respecting the number of available racks is ensured by adding flow constraints to the formulation. To express the quantity $R_i$, we have introduced dummy jobs (as many as racks) of duration 0, with the special feature that they do not release any flow at their completion. Thus the number of available racks decreases by 1 each time a dummy job is scheduled. Finally some simple cuts have been added to break the symmetries between batches of the same duration.

Weights $\alpha$ and $\beta$ of the two criteria have been chosen to reflect the preferences of the industrial user, evaluated on a set of schedules. Figure 2 represents an (optimal) initial schedule for an instance composed of 10 batches of duration 7 days, 10 batches of duration 14 days and 2 batches of 3 days. Here, weights are set equal to $\alpha = 0$ and $\beta = 1$, *i.e.* only the resource utilization is considered. The schedule starts on a Monday. Unavailability periods are represented on the Gantt chart by vertical dashed time intervals. In addition to weekends, several days-off appear. On top of the diagram, the occupation of other resources for filling operations is displayed (each case corresponds to half a day). The makespan of the schedule is 40 days and the average value of $R_i$ is 29.9 days.

Figure 3 shows the Gantt chart for the same instance with weights 1 and 5. The schedule completes after 36 days and the average value of $R_i$ is 30 days.

After extensive experimentation, this last parameter setting was finally adopted by the user.

#### 2.2.2 Main Schedule

After the initialisation procedure, all jobs are correctly started and a large number of new experiments are available for scheduling. Now, the system starts the main stage using a list schedule. The main idea of the algorithm is to share the resources (the racks) dynamically according to two criteria:

- a part of the reactors are allocated to the "late" jobs (advancing Phase I);
- the remaining reactors are allocated to the "earliest" jobs (finishing Phase II).

The first part ensures that enough experiments are generated: processing jobs in Phase I allow some chemicals to start their Phase II and hence generate new experiments. This ensures that the occupation rate of the racks remains reasonable. It also helps to minimize the makespan by decreasing the critical paths of the jobs.

The second part of the reactors is used to finish as rapidly as possible some jobs. This corresponds to the minimization of the total completion time $\Sigma C_j$ and allows to obtain regularly some results that can be used by the chemists. Indeed, they need all the results of the experiments for a given chemical to be able to assess a possible future exploitation.

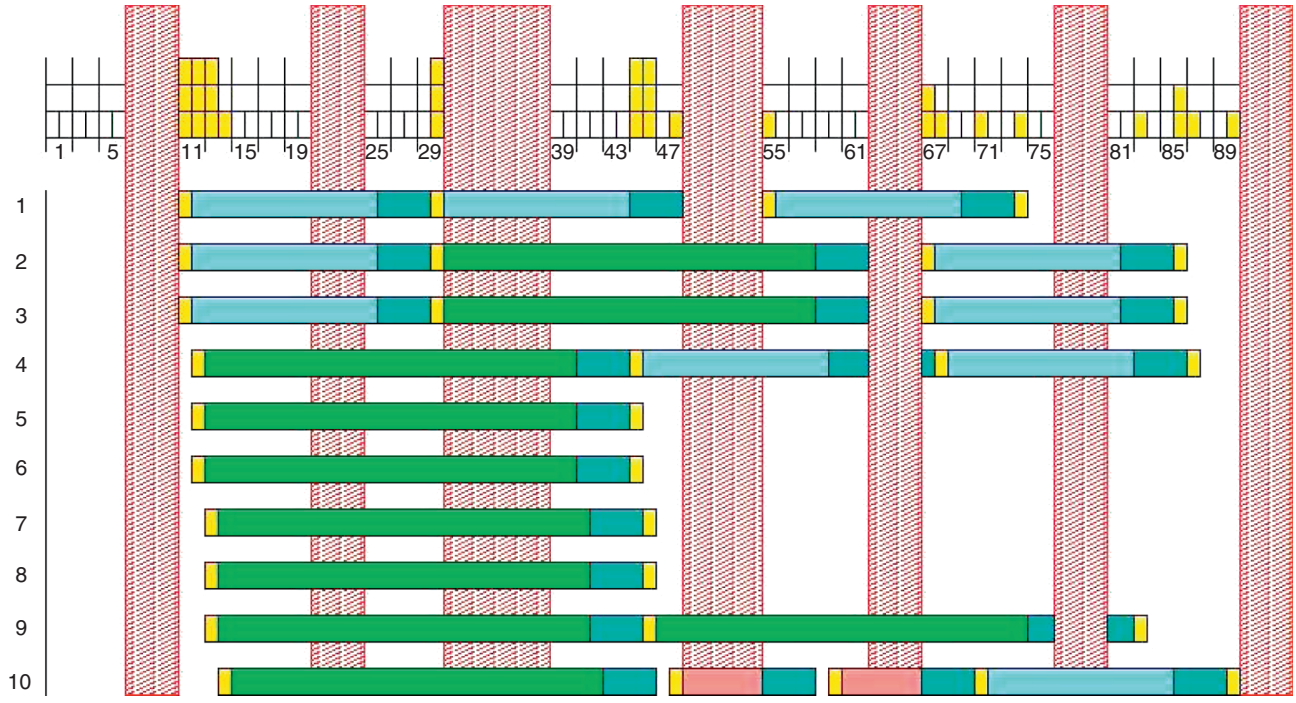


Figure 2

Optimal schedule for $\alpha = 0$ and $\beta = 1$. With these parameters, we only deal with resources criteria (*i.e.* racks must be free as soon as possible; the makespan is not a criteria). It takes then 40 days to carry out the first phase but 4 racks are available after 22 days.

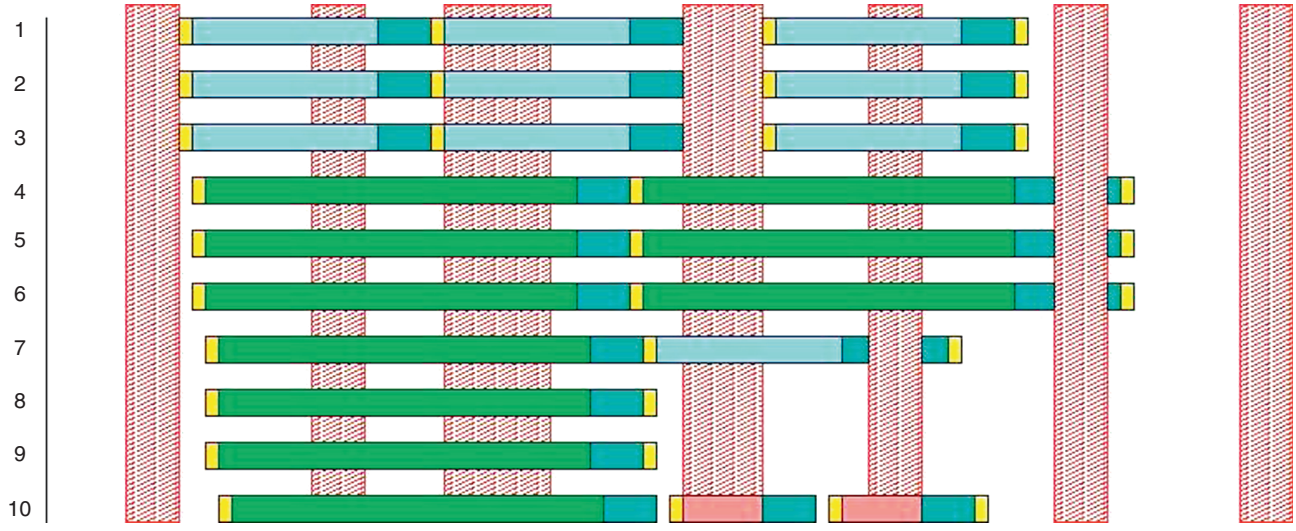


Figure 3

Optimal schedule for $\alpha = 1$ and $\beta = 5$. With these parameters, two criteria are used: resources criteria (*i.e.* racks must be free as soon as possible) and makespan (global time to carry out all experiments). It takes then 36 days to carry out the first phase but only 2 racks are available after 22 days.

TABLE 1
Algorithm

*At each instant when a rack is to be filled and is not pre-allocated:*
*calculate p**
*calculate the orders $\leq_A$ and $\leq_B$*
*if p* > p then*
*the first job in $\leq_B$ is allocated to the rack*
*else*
*choose a job among the first ones of $\leq_A$ and fix its schedule*
*update p according to the chosen job.*

The choice of the experiments to be placed in the reactors is done by a local optimisation according to the corresponding criterion. Some other practical constraints that cannot be described here because of confidentiality reasons are also considered in this choice and help improve the filling rate of the racks.

Before presenting the algorithm, we describe the notations. Each of the steps of the algorithm is then given in detail.

The set A corresponds to the early jobs, already in Phase II, that we would like to finish rapidly. The set B corresponds to the late jobs, still in Phase I, that the algorithm (*Tab. 1*) should advance.

Define on the set of all jobs $A \cup B$ the total orders $\leq_A$ and $\leq_B$ where the order $\leq_A$ favours the jobs of set $A$ and the order $\leq_B$ favours the jobs of set $B$.

Denote by $p$ the number of racks allocated to the progress of Phase I at a given instant. It is updated at each beginning or ending of a heating operation. $p^*$ is the "ideal" value of $p$.

We now describe in more detail each step of this algorithm.

*Calculate the orders $\leq_A$ and $\leq_B$*
The orders $\leq_A$ and $\leq_B$ reflect the priorities (according to one of the two criteria) that we want to give to the jobs. They are defined as follows:
– *order* $\leq_A$: the jobs with no operation left in Phase I, by increasing order of the length of the largest remaining operation. Then all jobs of $B$ follow;
– *order* $\leq_B$: the jobs that still have some experiments left in Phase I, by decreasing order of the length of their critical path. Then add the jobs of $A$.

It is not necessarily the first job of one of the two orderings that is chosen by the algorithm since one wants to maintain a high occupation rate of the racks.

*Calculate p**
$p$ is the current number of racks allocated to the progress of Phase I and $p^*$ is the ideal number of racks allocated to Phase I. The idea is to share in a fair manner the racks among the two sets $A$ and $B$ according to the remaining expected durations of each of them.

If $t_1$ is the expected time to finish the jobs of Phase I and $t_2$ is the expected time to finish the jobs of Phase II (with all the racks available), then we choose the following value of $p^*$ such that the two schedules have the same duration when sharing the $m$ racks:

$$p^* = m\frac{t_1}{t_1 + t_2}$$

*Choose a job and fix its schedule*
The objective is to choose the job that might finish the earliest. The candidates are known: the late jobs still in Phase I have only a small chance to be chosen. The choice will be made on jobs in Phase II for which we know all the remaining experiments. We can calculate the earliest completion time for each job by determining a schedule that minimizes its makespan $C_{\max}$.

This problem is hard in the general case but can be solved efficiently if there are not too many experiments and a small number of different values for the durations. Nevertheless, the calculation times can be prohibitive if we consider all the jobs of $A$: the order $\leq_A$ allows to limit the calculation time by restricting the search only to the first jobs:
– choose the job $a$: among the first jobs of $\leq_A$, choose the one that can end the most rapidly (taking into account the availability of the racks and reserving $p$ racks for Phase I) by calculating a schedule that minimizes the makespan $C_{\max}$;
– pre-allocation of the racks: fix the starting time of each remaining experiment of job $a$.

### 2.2.3 Implementation of Informatics System

We implemented the software for planning the experiments on a Pentium 4, CPU 2.53 GHz, RAM 1Go in Java 1.5. The interface is designed to enable the chemists to access all information concerning the incoming tasks (starting time, exact content of racks, duration) as well as advanced functionalities. Figure 4 shows the main window of the software [32].

The interface is in French as required by the user. For confidentiality reasons, the content of the racks is not displayed but it would appear under "*Informations sur la tâche courante*" (Information on the selected job). The diagram shows the different actions the chemists have to perform between December 11 and 18. "*Chargement*" stands for filling, "*Au four*" for heating, "*Filtration*" for filtering and "*Lavage*" for rinsing. "*Panne*" corresponds to a breakdown of some of the racks (the chemists are at the time unable to use the specified racks).

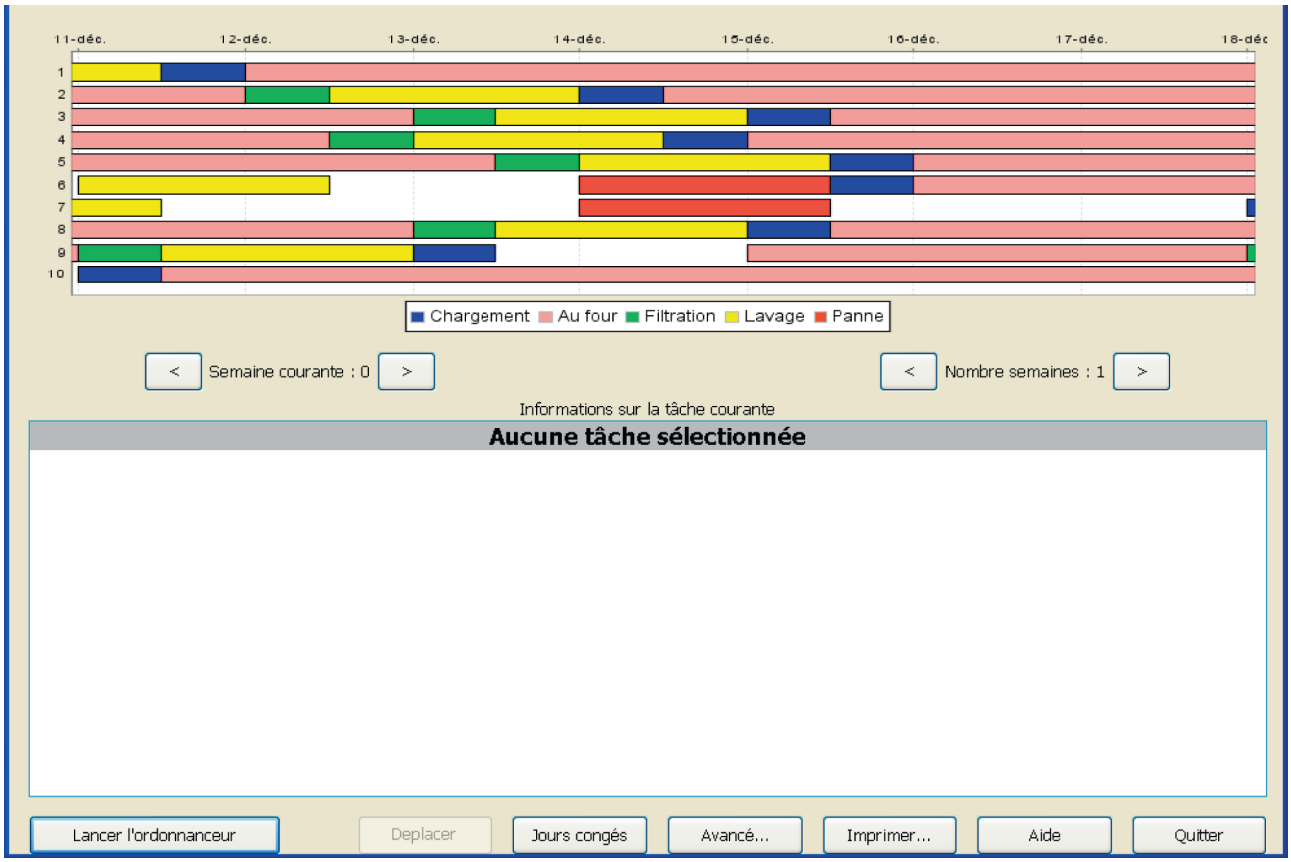


Figure 4

Main window of the software.

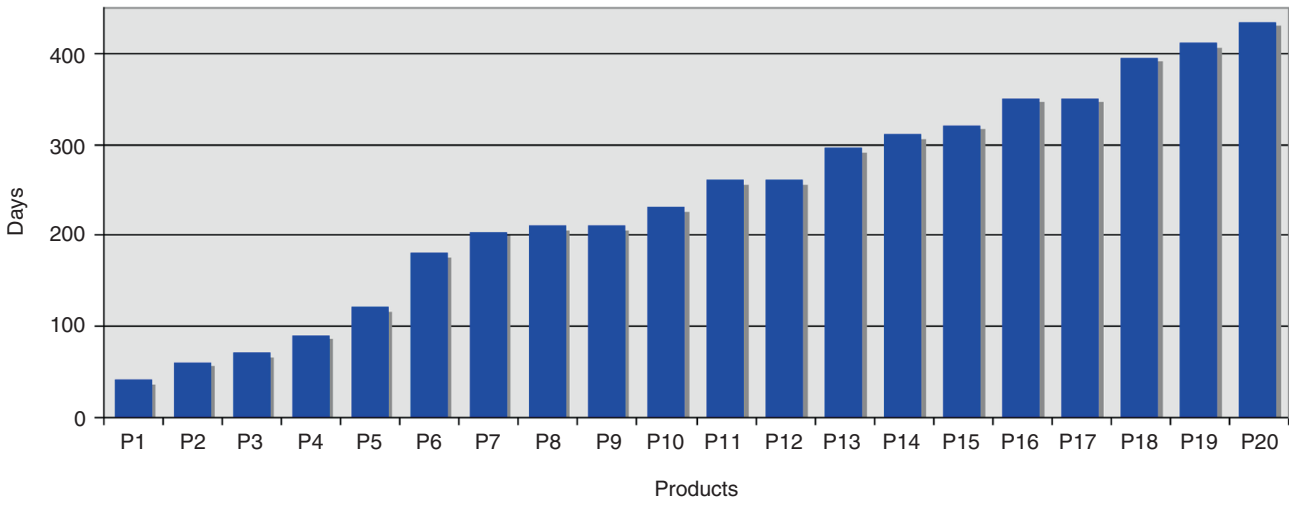


Figure 5

Finishing times of the chemicals (*y* axis: total time of the schedule for each product).

Figure 5 shows the planning of only one week but it is possible to increase the number of weeks appearing in the window to get an overall view of the schedule ("*Nombre de semaines*" button).

The software also allows to move a loading operation in case of an unexpected unavailability of the chemists ("*Déplacer*" button) and to add or remove days-off and holidays in the planning ("*Jour de congés*" button). The button "*Avancé*" offers advanced functionalities such as machine breakdown management, addition of new chemicals or cancelling operations (in case certain operations should be restarted for some reason). While using the interface, the chemists will be asked automatically for all the results of the experiments that have been obtained but have not yet been given to the software and plan the following actions accordingly.

None of these running times take more than fifteen minutes, even if the whole schedule has to be recalculated in order to take into account recent events (such as machine breakdowns or modifications of the days-off planning). This execution time has been well accepted by the laboratory.

### 2.3 Numerical Results

Due to confidentiality reasons, only numerical results will be presented. They were conducted to evaluate the performance of the scheduling algorithm. The size of the problem is too big to obtain optimal solutions for any of the objectives considered (prohibitive calculation times). However, the industrial objectives (the maximal filling of the racks, a regular output of the chemicals and the total time to finish the first 20 chemicals) were clearly defined and were attained. Below are some numerical results.

Figure 5 describes the finishing times of the chemicals. Those times are rather regular: a product is completely finished every 10 days on average with a standard deviation of 7.5 days. The total time to get the first 20 products is slightly greater than 200 days.

One of the industrial requirements was to fill the racks close to their capacity and not to let them unused. On the data set we had, the racks were used 90% of the time and filled on the average at 82%. Those numbers satisfied the industrial demand.

Figures 6 and 7 indicate the influence of the number of racks on the solution (average deviation between two products and total time of the schedule). Note that increasing the number of racks significantly reduces the times for both criteria. This shows that, despite Graham's anomalies that may appear in list algorithms, our approach uses efficiently the additional resources.

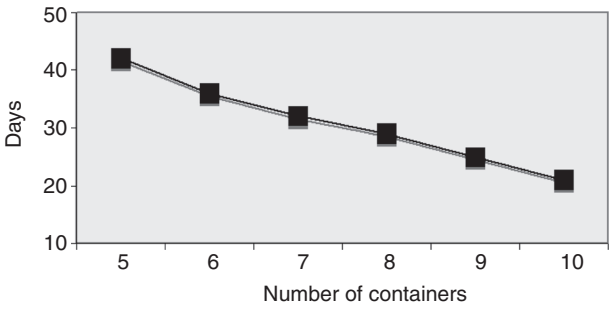


Figure 6

Mean deviation (in days) between the finishing times of two chemicals (in half days) depending on the number of racks available.

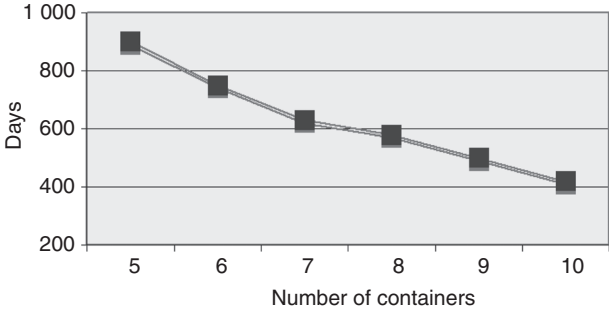


Figure 7

Makespan $C_{max}$ (in half days) depending on the number of racks available ($y$ axis: Makespan in days).

### 2.4 Conclusion

A software has been developed and implemented in Java, which concerns the scheduling of chemical experiments. The numerical tests carried out have shown that all objectives, required by the chemists/chemical engineer, have been satisfied: full utilization of the resources (racks), regular output of the final results for the different chemicals, and overall acceptable duration for the whole project. The program is actually used and gives full satisfaction to the final users.

## 3 DATA STORAGE AND MANAGEMENT

This section deals with data storage and management. It is not dedicated to one process. It is used even for zeolithe discovery (see previous sections) or other applications (reforming, hydrotreatment, etc.).

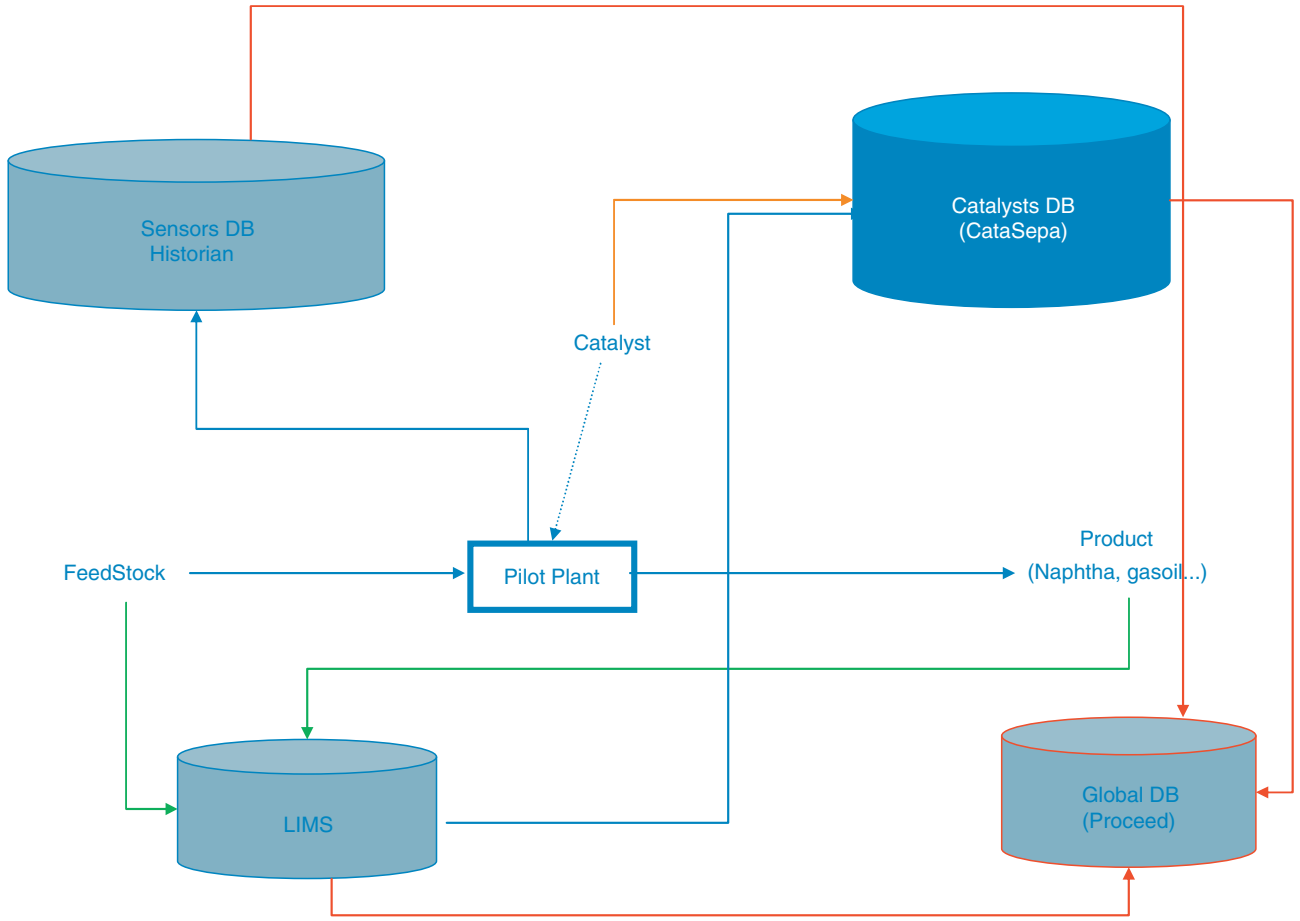


Figure 8
IFPEN global architecture.

### 3.1 Objectives

The number of data handled by an HTE methodology is relatively large (several tens of millions per year). It has therefore been necessary to create databases allowing long-term storage and easy processing of the data.

Commercial tools specific to HTE (pipeline pilot from Accelrys, etc.) were difficult to use since the aim was to integrate HTE tools as a standard pilot plant in the global Information System.

Similarly, the university platforms resulting from global project: CombiCat (catalyst design and optimisation by fast combinatorial analysis, [33]), TOPCOMBI (towards optimised chemical processes and new materials discovery by combinatorial science [34] are too specific to be used (see [35-38] for global review)).

We then developed databases based either on proprietary development (Java J2EE developments [39, 40]) according to standard MVC [41-43] or off-the-shelf tools (Teexma [44]). It complies with market standards. Data handling remains the difficult aspect.

This paragraph presents the global information system and how the problems to store and query data have been tackled.

### 3.2 Description of the Global IFP Energies Nouvelles Information System

We develop a global information system (either for HTE and conventional pilot plants) based on several databases (*Fig. 8*) [45]:

- a real-time database (Historian) to manage the sensors (around 15 000 sensors are managed);
- a Laboratory Information Management System (LIMS) database (*via* software package SQL*LIMS) to manage gas, liquid and solid samples analyses;

– a catalyst database (CataSepa) produced *via* a specific development based on standard J2EE to manage catalysts information (around 100 000 catalysts are stored with complete characterization);
– a process database (Proceed) to manage Process data coming from pilot plant (HTE or conventional) (around 200 pilot plants are managed, which corresponds to about 1 000 000 balances).

The latter database was produced using TEEXMA software, developed by Bassetti [44]. This configurable and scalable software is dedicated to collaborative management of technical and scientific data. It allows capitalisation but especially valorisation of technical knowledge. It is very well adapted to research centers.

All these databases have a bidirectional link (input and output) with Excel which is the main tool of chemical engineers.

### 3.3 Importance of Queries

Chemical repositories are complex. They involve hundreds of tables in numerous database instances. Access to the relevant information may be the result of business rules depending on data statuses (pending, validated, suspicious, etc.) or data source (*e.g.* analytical method for a content). Lastly, since research engineers constantly explore new avenues, the models are never frozen and must evolve dynamically. They are all modelled as meta-models or ontologies (models which describe themselves) in order to store new types of information by simple configuration.

This extreme storage flexibility, however, makes the data extremely difficult to access. The standard query models *via* data warehouses and data marts [9, 46, 47] that are widely found in the finance sector cannot be implemented with scalable self-described models. Consequently, traditional intelligence business tools (requesters, OnLine Analytical Processing (OLAP) cubes) cannot be used since they only apply to a fixed data model.

The intelligence must therefore be built into the modules and data access functions provided in our framework. Each data silo has its own high-level functions. They provide easy access to a single datum (catalyst DRT, sulphur content of a cut, test start date, etc.). They can be used with no knowledge of the underlying data model or specific business rules (data status, data priority). They hide the complexity of the repositories, thereby allowing flexible access to data. Consequently, there is no need for a dedicated datamart with creation of reports by computer specialists for each new requirement.

The user has then a set of dedicated functions providing access to centralised data (unitary, search, report) which, combined with the database insertion functions from Excel, can be used to analyse data from a single tool: Excel.

### 3.4 Data Analysis

Simple data can therefore be easily analysed in Excel. For multi-dimensional data, however, the use of a spreadsheet is sometimes limiting. One can use dedicated software (Matlab, Scilab, SAS, Sptofire, etc.). However in order to optimize productivity, we have decided to develop our own/proprietary dedicated software program (called Visu3D) to visualise data, explore data and build quickly correlation or models. This tool uses visual analytics concepts.

Visual analytics is a new domain. It differs from ordinary visualization: the active role is played by the computer in the presentation of information to the viewer. For the first time, we have a marriage of analytic statistical algorithms and visual presentation in an interactive environment. Before visual analytics, exploring high dimensional data with widgets like rotation controls, slice-and-dice tools, filter sliders, lensing tools and real-time brushes was a haphazard enterprise. Exploring raw high-dimensional data with such tools necessarily falls prey to the curse of dimensionality.

By contrast, visual analytics offers the prospect of guided exploration. Given interactive tools and underlying analytic components, a user can explore views of high-dimensional data that are highlighted by statistical algorithms. The result is the blending of the strengths of each approach: the analytic spotlight of statistical models and the inferential floodlight of visual exploration [48-52].

This tool is used for several principal purposes:
– checking raw data for anomalies. Anomalies in raw data include outliers caused by coding errors, sensor malfunctions, extreme environmental conditions and other factors. Anomalies also include missing values, which may occur randomly or deterministically. Eventually, anomalies may include biases due to response sets, ceiling and floor effects, and history and maturation effects. These biases can affect the shape of distributions assumed in data analysis;
– exploring data to discover plausible models. We call this Exploratory Data Analysis (EDA) [53]. EDA is not a fishing expedition. We explore data with expectations. We revise our expectations based on what we see in the data. And we iterate this process;
– elaborating model and correlation in order to predict some quantities;
– checking model assumptions. Checking model assumptions requires plotting residuals and other

diagnostic measures. These plots are often specialized, in order to assess distributional assumptions. In particular, surface plots are useful to test models built using DOE.

All of these tasks are well-documented in the statistics literature. Outliers, missing data and other anomalies are covered in [54-56]. EDA is discussed in [57-59]. Model diagnostics are presented in [60-63].

Some commercial tools are available (XLSTAT-3DPLOT, OriginLab, TableCurve 3D, Surfer 8, SpotFire, Gnuplot). However, it appears that these tools are not sufficient in order to extract the whole information contained in the data. In particular, none of these tools allow easily to visualize data points, to check data points, to classify data points and to build and check models.

Visu3D has hence been developed. It helps to visualize, build models and support the explorative screening of data specific to chemical industry. This tool reads data coming from Excel. It is inspired by recent works on that subject [48, 50, 64, 65]. Following paragraphs describe the main functionalities of the software.

### 3.4.1 Interactive Exploration Tools

This functionality allows the user to display multidimensional data, stored in an Excel file, as scatter plots. The studied data set is composed of $N$ experiments (lines of the Excel sheet) and $M$ variables (column of the Excel sheet). A column of the Excel sheet is called a dimension.

Up to six dimensions of the data set can be displayed on a graph. They are represented by:

- spatial coordinates ($X$, $Y$, $Z$),
- point colour,
- point symbol,
- point label.

One important point to well visualize complex data sets is the ability to explore and modify graphics interactively, for example (*Fig. 9*):

- select some points in a scatter plot to highlight the corresponding lines in the Excel sheet;
- select some lines in the Excel sheet to highlight the corresponding points in the scatter plots;
- suppress some outliers;
- rotate, zoom, filter, etc.

### 3.4.2 Principal Component Analysis and Classification

Visu3D can be used to analyze data sets including a lot of variables by principal component analysis and classification techniques.

Principal Component Analysis (PCA) [66, 67] can be applied on the whole or on a selected part of the data. Clicking on one point (which is the projection of one sample in the proper space) it is possible to see all the information relative to this sample.

Two classification techniques can also be applied:

- K-means algorithm [68]: its consists in choosing $K$ points randomly and then to aggregate the other points considering some specific distance (Mahalanobis distance) (*Fig. 10*).
- Clustering around Latent Variables (CLV): its strategy basically consists in performing a hierarchical cluster analysis, followed by a partitioning algorithm [67, 69] (*Fig. 11*).

### 3.4.3 Surface Response Visualization (*Fig. 12*)

Once the model has been computed, it is very helpful to visualize the surface response. In statistics, Response

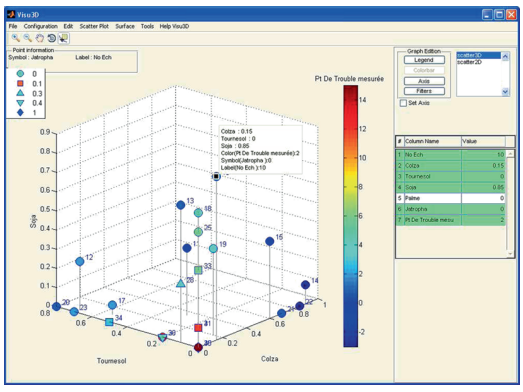


Figure 9

Six-dimension scatter plot.

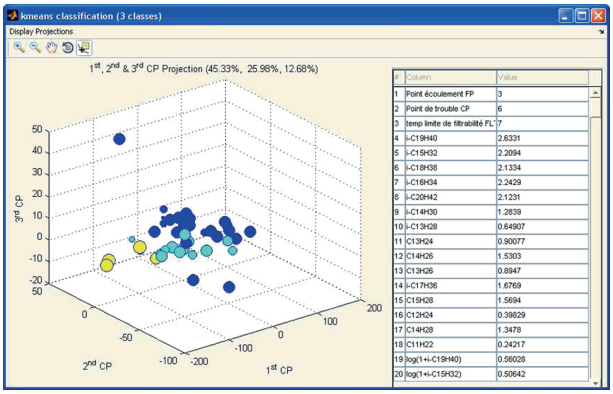


Figure 10

Example of K-means classification and PCA. By clicking on one point, user can see all information dedicated to this sample.

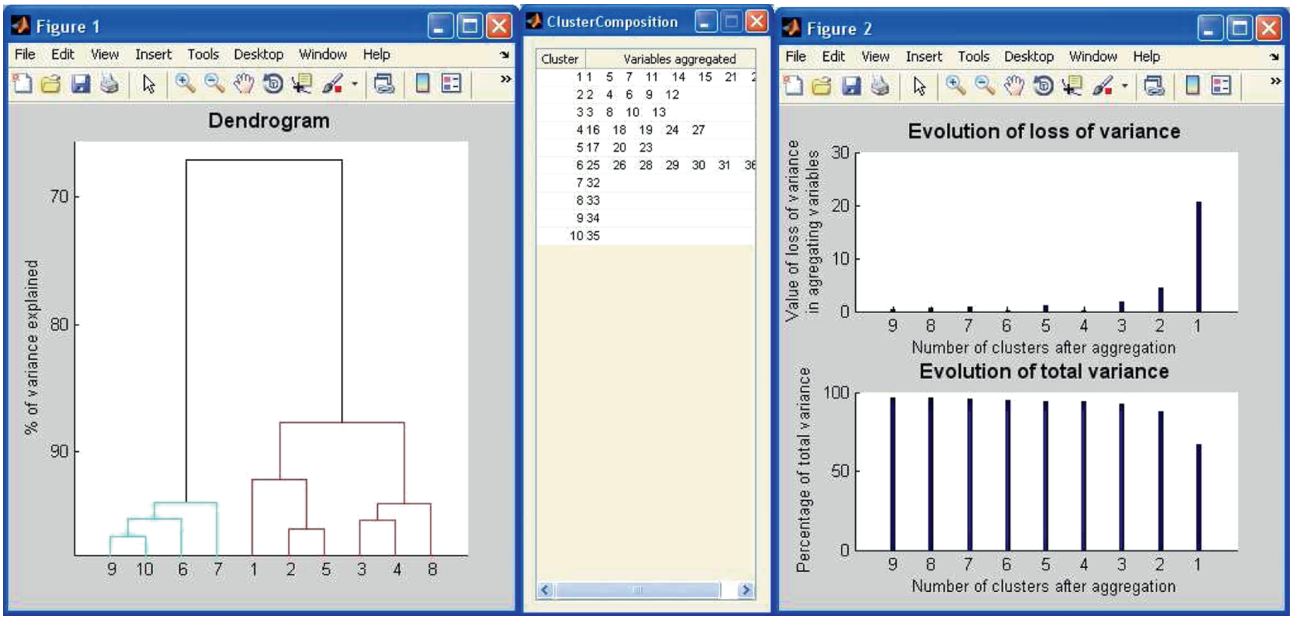


Figure 11

Windows obtained by CLV classification on data. Clustering analysis is then straight forward.

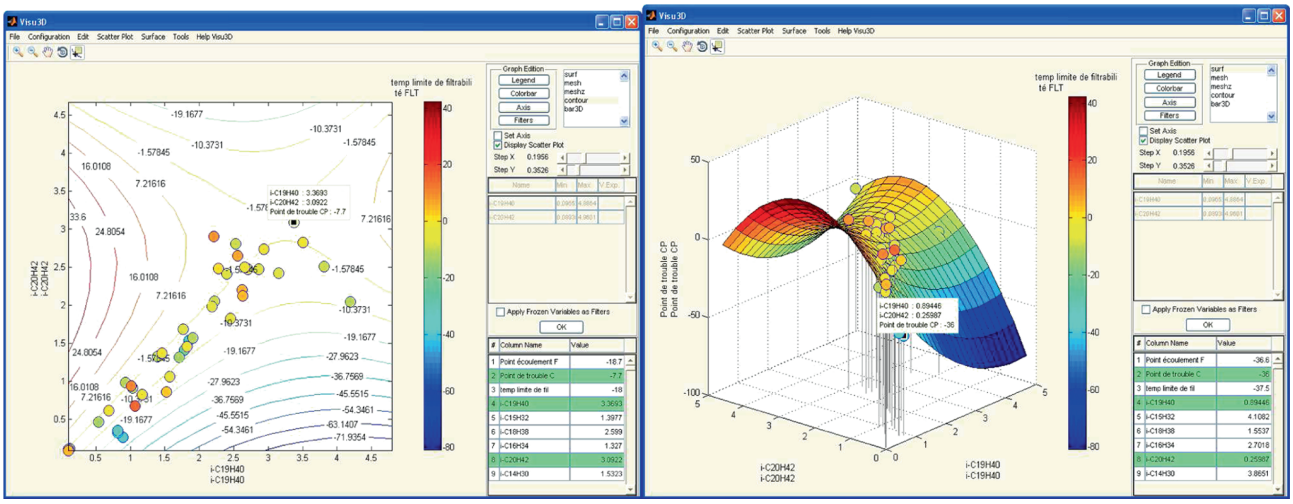


Figure 12

Nonlinear response surface and the associated scatter plot. a) Contour plot; b) surface plot.

Surface Methodology (RSM) explores the relationships between several explanatory variables and one or more response variables. The main idea of RSM is to use a sequence of designed experiments to obtain an optimal response. Box and Wilson suggest using a second-degree polynomial model to do this. They acknowledge that this model is only an approximation but use it because such a model is easy to estimate and apply, even when little is known about the process.

The user has still a lot of interactive exploration and visualization options: the surface has different ways of displaying, the size of the mesh is adjustable, the corre-

sponding scatter plot can easily be displayed and the user can also truncate the axis, change the legend, the title, etc.

#### 3.4.4 Parameter Estimation

An important functionality of Visu3D is to compute parameter estimation from a set of samples and some variables. Two possibilities are available:

- parameter-constrained models: given the data matrix **X** (where the variables are the columns and the samples are the rows) and a response vector $y$, the aim is to compute a vector of coefficients which minimizes the least square errors under some constraints. A trust-region-reflective algorithm was used. This technique is adapted to such nonlinear optimisation problems [57];
- nonlinear parametric models: some models, particularly in chemical science, are fundamentally nonlinear. To determine such nonlinear responses, the user must provide a parametric form. Then, Visu3D computes the coefficients using a standard Levenberg-Marquardt least squares algorithm [60, 61, 70].

#### 3.4.5 Exhaustive Selection Through Leaps and Bounds

The main problem in modeling is to estimate the influent variables. Classical methods (like Stepwise) do not test all the possible models, so the final selection is not the best one. However, testing all the possibilities would take too much time. In Visu3D, a leaped and bound algorithm has been adapted. It does not test all the possible selections but only a part of them that is sufficient to find the best one.

A regression graph to enumerate and evaluate all possible subset regression models is introduced. The graph is a generalization of a regression tree. All the spanning trees of the graph are minimum spanning trees and provide an optimal computational procedure for generating all possible submodels. Each minimum spanning tree has a different structure and characteristics. An adaptation of a branch-and-bound algorithm which computes the best-subset models using the regression graph framework is applied [71].

For example, Figure 13 shows the best model selection to predict pour trouble depending on concentration sample. Several models are tested.

This figure shows that in our example, 3 variables is the best compromise between precision and robustness [72]. Using this tool, correlation elaboration is then straight forward. Chemical engineer can quickly have access to the formula and check it comparing model with real data.

## 4 PILOT PLANT AUTOMATION EXAMPLES

A multi-reactor pilot plant offers considerable time savings *via* simultaneous synthesis of materials or tests. However, these time savings must not be lost by long, tedious manual parameter inputs which are a source of errors. Acquisition/development of multi-reactor and HTE tools must therefore be accompanied by the creation of simple and user-friendly input interfaces allowing the tools to be integrated in a global Workflow.

This chapter describes several practical applications to illustrate previous paragraphs:

- an Excel application to configure and monitor a catalyst synthesis tool;
- integration of an HTE pilot plant in a global environment with security constraints;
- monitoring pilot plants using Excel and Visu3D;
- easy multi-run comparison using Visu3D and multi pilot plants tests run comparison.

### 4.1 Excel Application to Configure and Monitor an HTE Pilot Plant

This paragraph describes an Excel application to configure and monitor an HTE pilot plant: a Synthesizer Large Trolley (SLT) developed by *Chemspeed Technologies*. This robot can be used to synthesise 16 catalysts in parallel (*Fig. 14*). It therefore offers considerable time savings: 16 catalysts are synthesised in 1 day (as compared with 8 days using standard methods on the bench).

The use of this robotic platform consists in performing a series of defined tasks, in a specific order: stirring, pH measurement, addition of products (solids or liquids) in defined quantities, etc.

Inputting the setpoint values and reading the saved values proved time-consuming and extremely repetitive. Storing information in databases can be also long and repetitive. These input tasks have therefore been automated by the development of an Excel interface. For each new series of syntheses, the interface is used to (*Fig. 15*):

- automatically save all the set point values in the pilot plant control software;
- retrieve the synthesis data stored in the pilot plant control software and display them for the user in an Excel sheet;
- save all these data in the dedicated database (CataSepa, *cf. Sect. 3.2*).

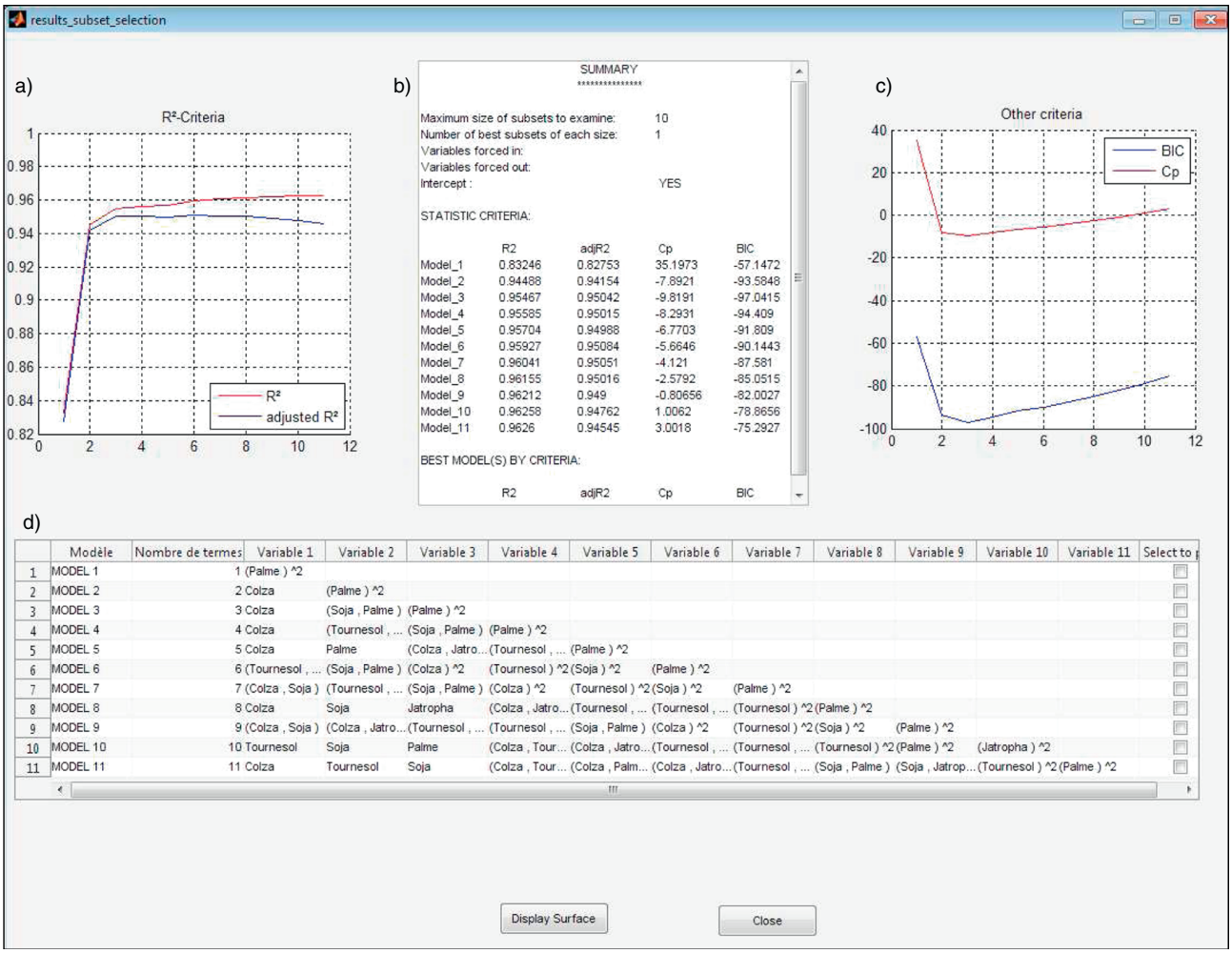


Figure 13

Example of best model selection. a) Variation of R2 depending on variables number used; b) different criteria to select the best model; c) variation of criteria depending on variables number; d) all models tested.


Figure 14

*Chemspeed* Accelerator SLT 100.

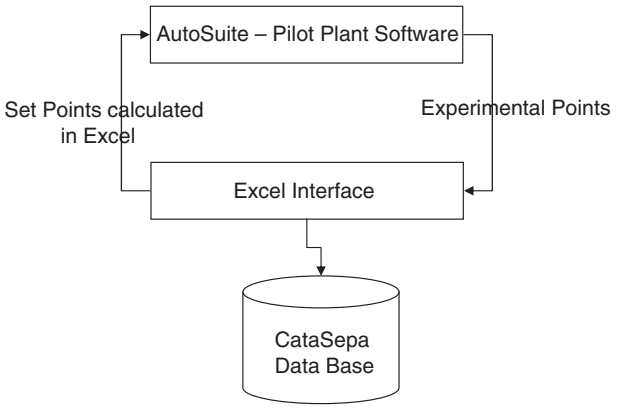


Figure 15

Features of the interface developed.

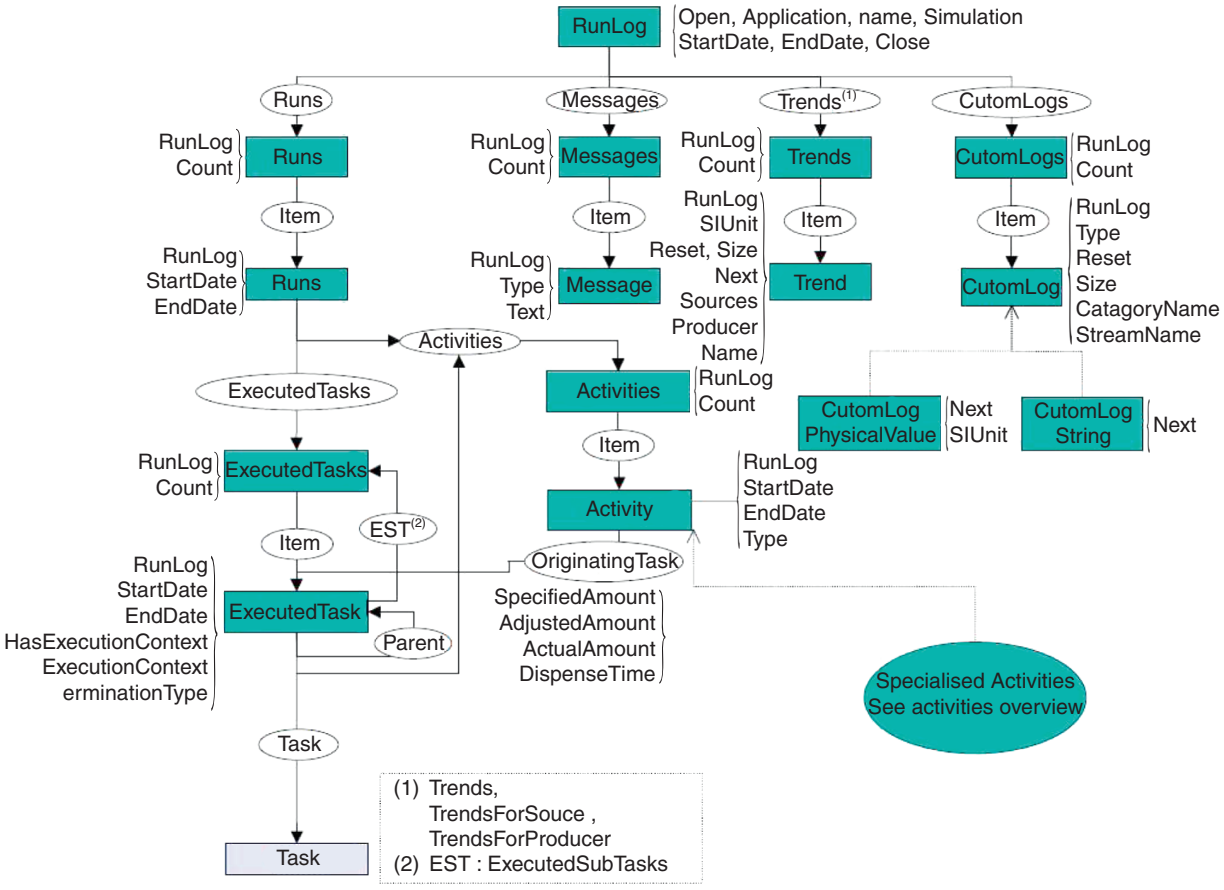


Figure 16

Overview of the structure of a RunLog object (representing the results of a synthesis).

This was carried out by a link between Excel and the pilot plant through a dedicated Application Programming Interface (API) provided by *ChemSpeed* (*Fig. 16*).

The storage of data (set points and results) in our data base dedicated to catalysts (CataSepa) is carried out using proprietary tools developed in our global framework (*cf. Sect. 3.2*).

This type of interface can be considered for all pilot plants. Similar interfaces have been developed for other robots of different brand (Tecan, Zinsser, etc.).

This interfacing limits retranscription errors and considerably reduces the overall times for transcription from one software program to another. For example, about 150 parameters must be transferred from an Excel file for a global preparation of 16 catalysts on a *Chemspeed* platform. Manual input would require about 2 hours. Input errors would be inevitable. With the current interface, it takes less than one minute. The risk of input error is reduced. For other transcriptions with different tools, the time of 5 to 6 hours necessary for manual input is reduced to a maximum of 1 hour with automated input.

### 4.2 Integration of an HTE Pilot Plant in a Global Environment with Security Constraints

When integrating a HTE pilot plant in a research centre, the "safety" aspects cannot be neglected. The operating mode must be globalized by using standard supervisors. This paragraph describes the adaptation of an Avantium Flowrence™ unit in the complete fleet of IFPEN pilot plants.

With this kind of pilot plant, the alarms generated are stored in each specific database. Then, they cannot be analysed by the global supervisor. A link of these specific data bases to our global hypervisor (FIX) has then been

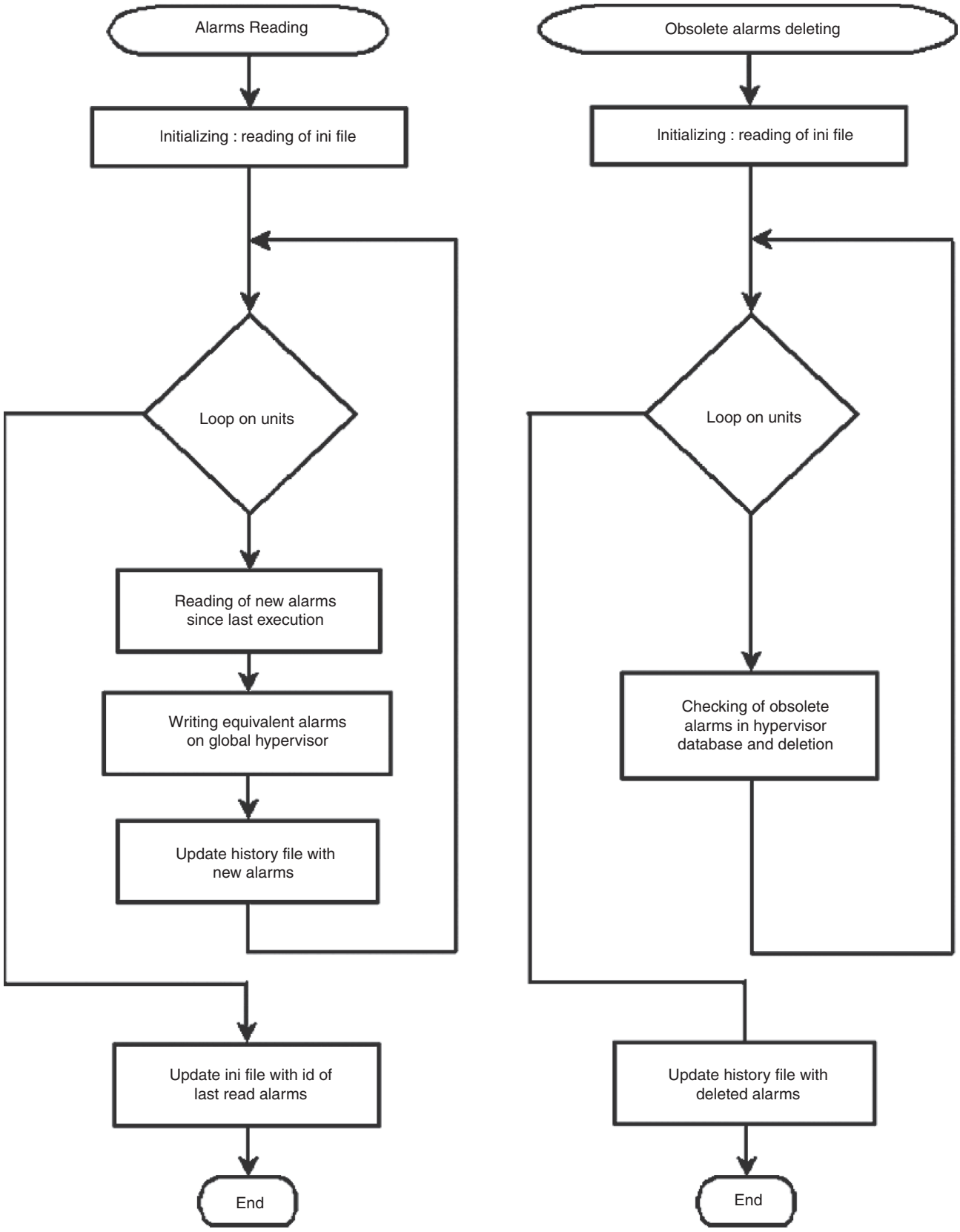


Figure 17

Processing principle.

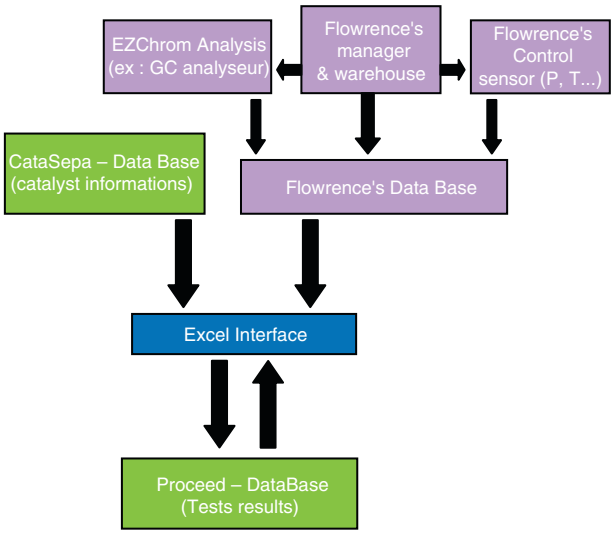


Figure 18

Proposed operating mode.

developed. Alarm can then be monitored by the global supervisor. It is not required to be in front of the pilot plant during experiments.

Two periodic processing operations have been developed:

– analysis of the alarm table to transfer the alarms to the hypervisor;
– purge of acknowledged alarms. The processing principle is described by the previous flowchart (*Fig. 17*).

### 4.3 Monitoring HTE Pilot Plant

This paragraph shows an example of a catalyst test monitoring. A special interface between the pilot plant (Flowrence from *Avantium*) and Excel has been developed using methods similar to the one presented in Section 4.1. All measures can be first visualized in Excel and then in Visu3D (cf. *Sect. 3.4*).

The following diagram describes the interfacing between the various software programs (*Fig. 18*).

At the end of each test run, a validation phase is carried out to define useful data and delete aberrant data (outlier). This will simplify multi-test and/or multi-reactor operation. For example, a standard display is the deviation between the process value of the sensors and the setpoints requested (*Fig. 19*). In this case, the test pressure is 59 bar with a maximum tolerated deviation of $\pm 0.5$ bar. Consequently, in this case, reactors 2, 11 and 16 cannot be used.

This task might be very tedious but with these dedicated tools it is straight forward.

### 4.4 Several Run Comparisons

Data visualisation can also be used to compare several test runs and to estimate repeatability of one experiment.

For example, a first screening of catalysts with respect to process data can be carried out:

– analysis of measurement repeatability (*Fig. 20* at the top and bottom left). This graph can be used to analyse the following points:
– repeatability on A- or B-conversion;
– mean and standard deviation of this conversion;
– can the catalysts be classified with respect to their level of A- and/or B-conversion?

Visu3D can be used to quickly:

– check if repeatability is below a threshold (the maximum deviation obtained is less than the calculated maximum tolerated deviation);
– compare catalysts.

In this example, Visu3D can provide several information:

– effect of operating parameters on a conversion (see *Fig. 20* at top right). In the proposed example, we can conclude that, in this pressure range, there is no effect, since at iso-conversion of A, for example between 56% and 60% for reactors 1, 5, 9 and 13, the pressure varies from 58.8 bar to 59.5 bar;
– analysis of B/A selectivity. In the example proposed (see *Fig. 20* at bottom right), we can conclude that the higher the A-conversion, the more selective the catalysts.

The test data can be subsequently analysed in combination with catalytic data, for example the area, porous volume, metal contents, etc. (see *Sect. 3.3*), and a 2D or 3D graph plotted.

Moreover, as all test runs are stored in our database (Proceed), and comparison between runs from HTE and conventional pilot plant is very easy. Figure 21 is a classical example to compare tests runs on several pilot plants.

## CONCLUSION

Data management is critical when setting up a HTE system. This paper presents development of several tools based on Excel (the favourite tool of chemical engineers) dedicated to HTE pilot plant. These tools allow to maximize productivity gains. They are gathered in a global

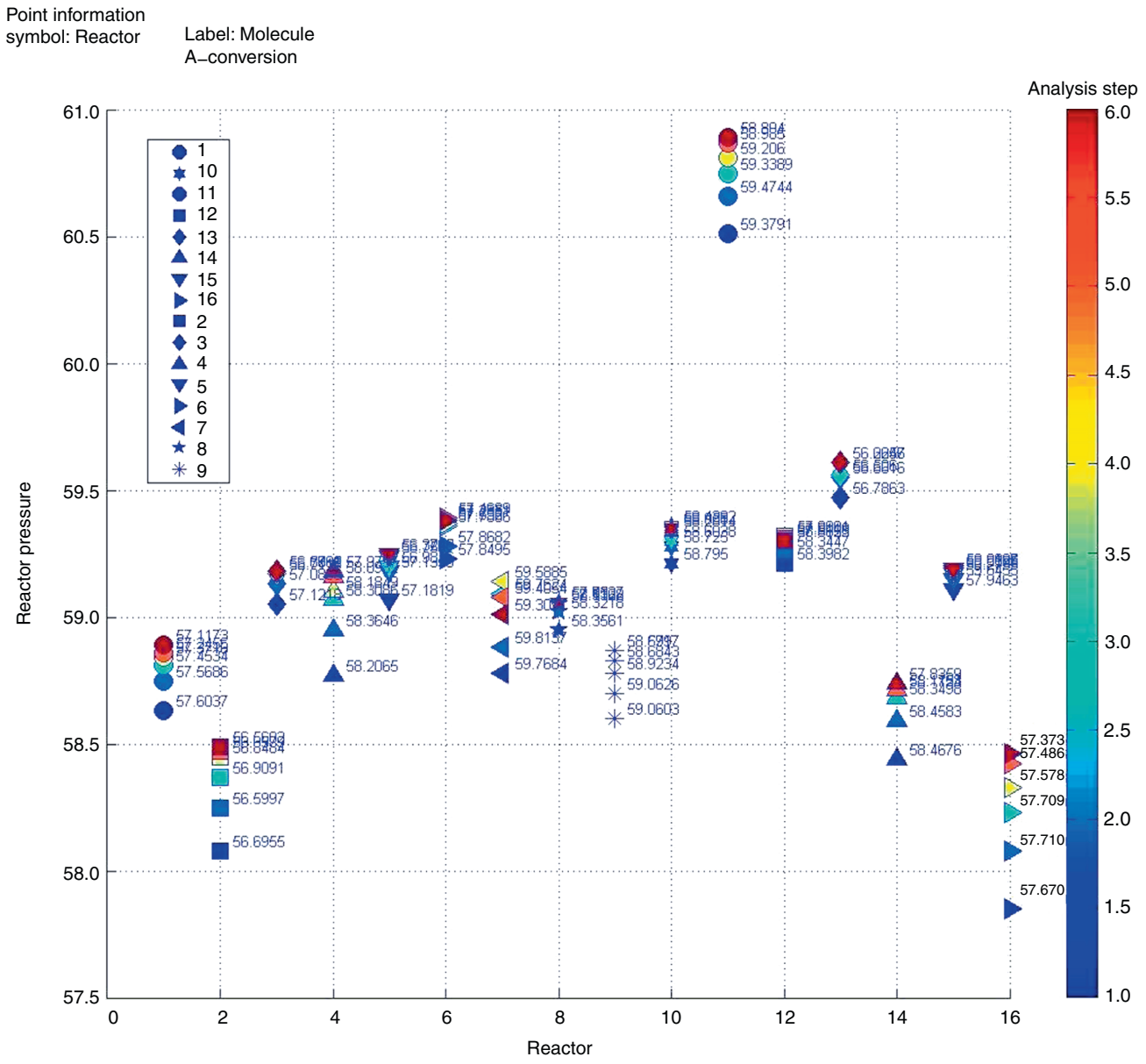


Figure 19

Example of multi-reactor representations.

framework dedicated to pilot plant management (conventional and HTE pilot plant). It includes:

– implementation of experiment planning tools. They are based either on the conventional DOE or on adapted production planning tools. These tools are extremely efficient, especially in the synthesis of zeolite which may require very long batch times (several weeks);
– databases. Storage in suitable relational databases, storage is simplified by suitable input interfaces (from Excel) avoiding long, tedious inputting of HTE unit parameters which may introduce errors. The productivity gain is quite impressive;
– the reporting aspects provide easy access to data. Users can easily access their data from Excel and can therefore retain their standard data processing tool;
– the "safety" aspects. Off-the-shelf HTE pilot plants are integrated in the global supervisor and are therefore monitored;

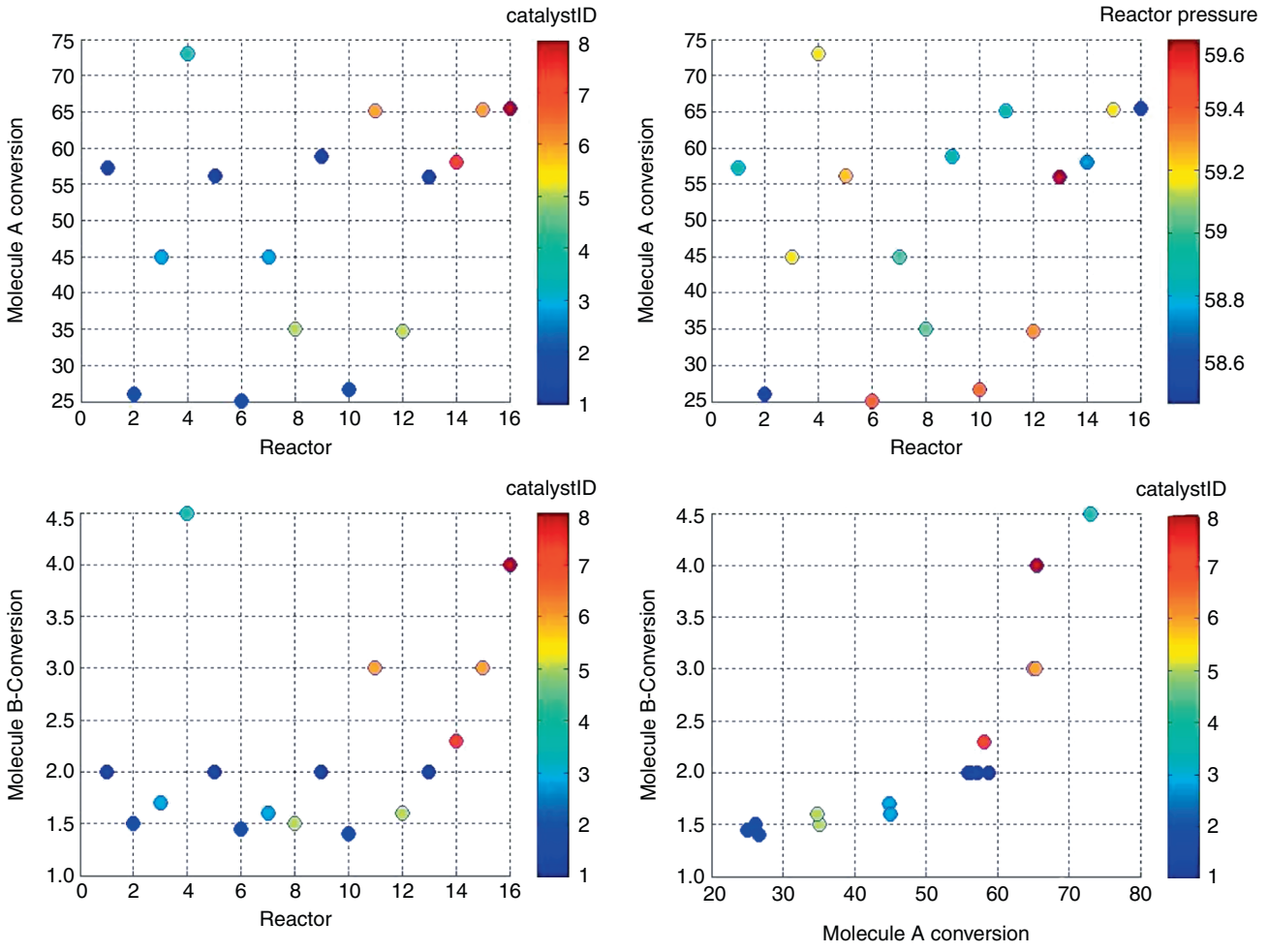


Figure 20

Example of inter-test comparison.

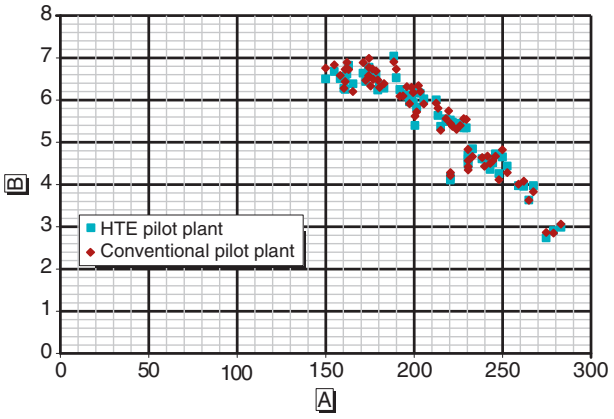


Figure 21

Comparison between HTE and conventional pilot plant.

– the "data exploration" and "data mining" aspects. A specific tool has been developed, to display data but also to produce models automatically (response surfaces, etc.). The tool can be used to determine optimum correlations according to several criteria (rR2, AIC, BIC, etc.), very quickly and very easily.

This framework allows efficient management of the conventional and HTE pilot plants. Others tools dedicated to optimise data processing have been developed [73-77].

Apart from the technical problems, it was set up through close collaboration between the computer specialists and the chemist engineers. The use of a dedicated methodology (Agile method) has been a key factor in the success of the project.

These tools allow the chemist engineer to focus on the data and therefore develop new catalysts or processes

more quickly and more efficiently. This is the key to success for integration of HTE pilot plants in the research centers.

## REFERENCES


1 Agile Alliance (2012) http://www.agilealliance.org/the-alliance/.

2 Meguro S., Ohnishi T., Lippmaa M., Koinuma H. (2005) Elements of informatics for combinatorial solid-state materials science, *Meas. Sci. Technol.* **16**, 1, 309-316.

3 Zhang W.H., Fasolka M.J., Karim A., Amis E.J. (2005) An informatics infrastructure for combinatorial and high-throughput materials research built on open source code, *Meas. Sci. Technol.* **16**, 1, 261-269.

4 Farrusseng D., Baumes L., Vauthey I., Hayaud C., Denton P., Mirodatos C. (2002) The Combinatorial Approach for Heterogeneous Catalysis: A Challenge for Academic Research, in *Principles and Methods for Accelerated Catalyst Design and Testing*, Springer, The Netherlands.

5 Derouane E. (2002) Principles and Methods for Accelerated Catalyst Design, Preparation, Testing, and Development: Conclusions of the Nato Advanced Study Institute, in *Principles and Methods for Accelerated Catalyst Design and Testing*, Springer, The Netherlands.

6 Ausfelder F., Baumes L.A., Farrusseng D. (2011) Preface, *Catal. Today* **159**, 1, 1.

7 Adams N., Schubert U.S. (2004) Software solutions for combinatorial and high-throughput materials and polymer research, *Macromol. Rapid Comm.* **25**, 1, 48-58.

8 *LabVIEW Graphical Instrument Control*, http://www.ni.com/.

9 Frantzen A., Sanders D., Scheidtmann J., Simon U., Maier W.F. (2005) A flexible database for combinatorial and high-throughput materials science, *QSAR Comb. Sci.* **24**, 1, 22-28.

10 Farrusseng D., Clerc F., Mirodatos C., Azam N., Gilardoni F., Thybaut J.W., Balasubramaniam P., Marin G.B. (2007) Development of an integrated informatics toolbox: HT kinetic and virtual screening, *Comb. Chem. High Throughput Screening* **10**, 2, 85-97.

11 Jiang J., Jorda J.L., Yu J., Baumes L.A., Mugnaioli E., Diaz-Cabanas M.J., Kolb U., Corma A. (2011) Synthesis and Structure Determination of the Hierarchical Meso-Microporous Zeolite ITQ-43, *Science* **333**, 6046, 1131-1134.

12 Fecant A. (2007) Synthesis of new zeolites with pore sizes of 10 and 12 tetrahedric atoms, PHD.

13 Baumes L.A., Moliner M., Corma A. (2007) Prediction of ITQ-21 zeolite phase crystallinity: Parametric versus non-parametric strategies, *QSAR Comb. Sci.* **26**, 2, 255-272.

14 Barr G., Dong W., Gilmore C.J. (2004) PolySNAP: a computer program for analysing high-throughput powder diffraction data, *J. Appl. Crystallogr.* **37**, 4, 658-664.

15 Baumes L.A., Kruger F., Jimenez S., Collet P., Corma A. (2011) Boosting theoretical zeolitic framework generation for the determination of new materials structures using GPU programming, *Phys. Chem. Chem. Phys.* **13**, 10, 4674-4678.

16 Deem M.W., Pophale R., Cheeseman P.A., Earl D.J. (2009) Computational Discovery of New Zeolite-Like Materials, *J. Phys. Chem. C* **113**, 51, 21353-21360.

17 Cawse J.N., Gazzola G., Packard N. (2011) Efficient discovery and optimization of complex high-throughput experiments, *Catal. Today* **159**, 1, 55-63.

18 Amanna A.E., Ali D., Fitch D.G., Reed J.H. (2012) Parametric optimization of software defined radio configurations using design of experiments, *Analog Integr. Circuits Signal Process.* **73**, 2, 637-648.

19 Kleijnen J.P.C. (2005) An overview of the design and analysis of simulation experiments for sensitivity analysis, *Eur. J. Operation. Res.* **164**, 2, 287-300.

20 Straetemans R., O'Brien T., Wouters L., Van Dun J., Janicot M., Bijnens L., Burzykowski T., Aerts M. (2005) Design and analysis of drug combination experiments, *Biom. J.* **47**, 3, 299-308.

21 King C.W. (2006) Statistics for experimenters, design, innovation and discovery, *AIChE J.* **52**, 7, 2657-2657.

22 Mazerolles G., Mathieu D., Phantanluu R., Siouffi A.M. (1989) Computer-Assisted Optimization with Nemrod Software, *J. Chromatogr.* **485**, 433-451.

23 Brucker P., Gladky A., Hoogeveen H., Kovalyov M.Y., Potts C., Tautenhahn T., Van De Velde S. (1998) Scheduling a batching machine, *J. Schedul.* **1**, 31-55.

24 Potts C.N., Kovalyov M.Y. (2000) Scheduling with batching: A review, *Eur. J. Operation. Res.* **120**, 2, 228-249.

25 Boudhar M., Finke G. (2000) Scheduling on a batch machine with job compatibilities, *Belgian J. Operations Res.* **40**, 69-80.

26 Brauner N., Finke G., Lehoux-Lebacque V., Rapine C., Kellerer H., Potts C., Strusevich V. (2009) Operator non-availability periods, *4OR-Q J. Oper. Res.* **7**, 3, 239-253.

27 Brauner N., Finke G., Lehoux-Lebacque V., Rapine C., Kellerer H., Potts C., Strusevich V. (2009) Operator non-availability periods, *4OR: A Quarterly Journal of Operations Research* **7**, 3, 239-253.

28 Rapine C., Brauner N., Finke G., Lebacque V. (2012) Single machine scheduling with small operator-non-availability periods, *J. Schedul.* **15**, 2, 127-139.

29 Schmidt G. (2000) Scheduling with limited machine availability, *Eur. J. Operational Res.* **121**, 1, 1-15.

30 Sanlaville E., Schmidt G. (1998) Machine scheduling with availability constraints, *Acta Informatica* **35**, 9, 795-811.

31 Blazewicz J., Ecker K., Pesch E., Schmidt G., Weglarz J. (2001) *Scheduling Computer and Manufacturing Processes*, 2nd ed., Springer-Verlag, Berlin, Heidelberg.

32 Lebacque V., Brauner N., Celse B., Finke G., Rapine C. (2007) Planification d'expériences dans l'industrie chimique, in *Les systèmes de production, applications interdisciplinaires et mutations*, Boujut J.-F., Llerena D., Brissaud D. (eds), Hermès Lavoisier, Paris.

33 Holzwarth A., Denton P., Zanthoff H., Mirodatos C. (2001) Combinatorial approaches to heterogeneous catalysis: strategies and perspectives for academic research, *Catal. Today* **67**, 4, 309-318.

34 TOPCOMBI (2012) www.topcombi.org.

35 Mills P.L., Quiram D.J., Ryley J.F. (2007) Microreactor technology and process miniaturization for catalytic reac-

tions perspective on recent developments and emerging technologies, *Chem. Eng. Sci.* **62**, 24, 6992-7010.

36 Corma A., Moliner M., Serra J.M., Serna P., Baumes L.A. (2006) A New Mapping/Exploration Approach for HT Synthesis of Zeolites, *Chem. Mater.* **18**, 14, 3287-3296.

37 Baumes L.A., Jimenez S., Corma A. (2011) hITeQ: A new workflow-based computing environment for streamlining discovery. Application in materials science, *Catal. Today* **159**, 1, 126-137.

38 Farrusseng D. (2008) High-throughput heterogeneous catalysis, *Surf. Sci. Reports* **63**, 11, 487-513.

39 J2EE Blueprints Digest, http://java.sun.com/developer/technicalArticles/J2EE/DesignEntApps/.

40 JavaTM Platform, Enterprise Edition 5 Specification, http://jcp.org/aboutJava/communityprocess/final/jsr244/.

41 Krasner G., Pope S. (1988) A cookbook for using the model-view controller user interface paradigm in Smalltalk-80, *J. Object Oriented Program.* **1**, 3, 26-49.

42 Reenskaug T. (2003) *The Model-View-Controller(MVC)Its Past and Present*, http://folk.uio.no/trygver/2003/javazonejaoo/MVC_pattern.pdf.

43 MVC, http://addyosmani.com/blog/understanding-mvc-and-mvp-for-javascript-and-backbone-developers/.

44 Teexma, www.bassetti.fr.

45 Cauvin S., Barbieux M., Carrie L., Celse B. (2008) A generic scientific information management system for process engineering, *18th European Symposium on Computer Aided Process Engineering, Comput. Aided Chem. Eng.* **25**, 931-936.

46 Ullman J.D. (1987) Database Theory: Past and Future, *Proceedings of the sixth ACM SIGACT-SIGMOD-SIGART Symposium on Principles of Database Systems*, San Diego, California, 23-25 March.

47 Vardi M.Y. (2000) Constraint satisfaction and database theory: a tutorial, *PODS '00: Proceedings of the nineteenth ACM SIGMOD-SIGACT-SIGART Symposium on Principles of Database Systems, ACM 2000*, Dallas, Texas, 15-17 May.

48 Purchase H.C., Andrienko N., Jankun-Kelly T.J., Ward M. (2008) Theoretical Foundations of Information Visualization, Kerren A., Stasko J.T., Fekete J.-D., North C. (eds), *Information Visualization*, Springer, Berlin, Heidelberg, *Lecture Notes Comput. Sci.* **4950**, 46-64.

49 Ji Soo Yi, Youn ah Kang, Stasko J., Jacko J. (2007) Toward a Deeper Understanding of the Role of Interaction in Information Visualization, *IEEE Trans. Vis. Comput. Graph.* **13**, 6, 1224-1231.

50 Keim D., Andrienko G., Fekete J.-D., Görg C., Kohlhammer J., Melançon G. (2008) Visual Analytics: Definition, Process, and Challenges, Kerren A., Stasko J.T., Fekete J.-D., North C. (eds), *Information Visualization*, Springer, Berlin, Heidelberg, Lecture Notes Comput. Sci **4950**, 154-175.

51 Shneiderman B. (2002) Inventing Discovery Tools: Combining Information Visualization with Data Mining? *IVS* **1**, 1, 5-12.

52 Lungu M., Xu K. (2007) Biomedical Information Visualization, Kerren A., Ebert A., Meyer J. (eds), *Human-Centered Visualization Environments*, Springer, Berlin, Heidelberg.

53 Tukey J.W. (1977) *Exploratory Data Analysis*, Addison-Wesley Publishing Company.

54 Young W.R. (1980) Outliers in Statistical Data, *Technometrics* **22**4, 631-631.

55 Bremer R. (1995) Outliers in Statistical Data, *Technometrics* **37**, 1, 117-118.

56 Rousseeuw P.J., Leroy A.M. (1987) References, in *Robust Regression and Outlier Detection*, John Wiley & Sons, New york.

57 Byrd R.H., Gilbert J.C., Nocedal J. (2000) A trust region method based on interior point techniques for nonlinear programming, *Math. Program.* **89**, 1, 149-185.

58 Brereton R.G. (2007) Pattern Recognition, in *Applied Chemometrics for Scientists*, John Wiley & Sons, Ltd, Chichester, UK.

59 Cook R.D. (1998) Introduction, in *Regression Graphics: Ideas for Studying Regressions Through Graphics*, John Wiley & Sons, Inc., Hoboken, NJ, USA.

60 Seber G.A.F., Wild C.J. (2005) Model Building, in *Nonlinear Regression*, John Wiley & Sons, Inc., Hoboken, NJ, USA.

61 Seber G.A.F., Wild C.J. (2005) Statistical Inference, in *Nonlinear Regression*, John Wiley & Sons, Inc., Hoboken, NJ, USA.

62 Seber G.A.F., Wild C.J. (2005) Errors-in-Variables Models, in *Nonlinear Regression*, John Wiley & Sons, Inc., Hoboken, NJ, USA.

63 Seber G.A.F., Wild C.J. (2005) Multiresponse Nonlinear Models, in *Nonlinear Regression*, John Wiley & Sons, Inc., Hoboken, NJ, USA.

64 Wilkinson L., Anushka A., Grossman R. (2006) High-Dimensional Visual Analytics: Interactive Exploration Guided by Pairwise Views of Point Distributions, *IEEE Trans. Vis. Comput. Graph.* **12**, 6, 1363-1372.

65 Fekete J.D. (2004) The InfoVis Toolkit, *IEEE_infovis, 10th IEEE Symposium on Information Visualization (InfoVis 2004)*, Austin, TX, 10-12 Oct., IEEE Press, pp. 167-174.

66 Ledauphin S., Hanafi M., Qannari E.M. (2004) Simplification and signification of principal components, *Chemom. Intell. Lab. Syst.* **74**, 2, 277-281.

67 Sahmer K., Vigneau E., Qannari E.M. (2006) A cluster approach to analyze preference data: Choice of the number of clusters, *Food Qual. Prefer.* **17**, 3-4, 257-265.

68 Seber G.A.F., Wild C.J. (1989) Wiley Series in Probability and Statistics, in *Nonlinear Regression*, John Wiley & Sons, Inc., Hoboken, NJ, USA.

69 Vigneau E., Qannari E.M. (2003) Clustering of Variables Around Latent Components, *Commun. Stat. Simul. Comput.* **32**, 4, 1131-1150.

70 Seber G.A.F., Wild C.J. (1989) Estimation Methods, in *Nonlinear Regression*, John Wiley & Sons, Inc., Hoboken, NJ, USA.

71 Gatu C., Yanev P.I., Kontoghiorghes E.J. (2007) A graph approach to generate all possible regression submodels, *Comput. Stat. Data Anal.* **52**, 2, 799-815.

72 Celse B. (2007) Reconnaissance de formes pour la conduite, in *Supervision des procédés complexes*, Gentil S. (ed.), Hermes Science Publications, Lavoisier, Paris.

73 Rouleau L., Celse B., Duchêne P., Llido E., Szymanski R. (2005) Multistage cross flow ion exchange process for zeolite: prediction method applied to MFI and MAZ, Proceedings of the *3rd International Zeolite Symposium (3rd FEZA), Stud. Surf. Sci. Catal.* **158**, 1105-1112.

74 Celse B., Bertoncini F., Duval L., Adam L. (2007) Automatic Template fit in comprehensive two dimensional gas chromatography images, Riva Del Garda, 1-1-2007.

75 Celse B., Bres S., Adam F., Bertoncini F., Duval L (2007) Polychrom: A Comprehensive GC*GC data handling software, *Gulf Coast Conference*, Houston, 1-1-2007. Galveston, Texas, USA, 16-17 Oct.

76 Celse B., Gueroult P., Moreaud F., Sorbier L. (2007) Determination of microscopic particle size using region growing and active contours: a pratical approach, *Reconnaissance des Formes et Intelligence Artificielle, RFIA Congress*, Reims, France, 1-1.

77 Ould-Chikh S., Celse B., Hemati M., Rouleau L. (2009) Methodology of mechanical characterization of coated spherical materials, *Powder Technol.* **190**, 1-2, 19-24.

*Manuscript accepted in December 2012*
*Published online in July 2013*